**Responses**

**RISPOSTA 1**

Il Database è un contenitore/repository di dati, insieme di tabelle, organizzati in maniera CONSISTENTE, nel senso che una base dati non dovrebbe contenere valori inconsistenti, ovvero errati o che non hanno motivo di esistere. Inoltre, deve essere garantita l’integrità referenziale del dato

**RISPOSTA 2**

Il DBMS (Database Management System) è un software che gestisce basi dati e consente di interrogarle. Consente anche di gestire gli accessi.

IL DBMS si può implementare On premise (in locale) oppure può essere installato anche sul Cloud.

Talvolta detto anche RDBMS (Relationship database management system) perchè gestisce basedati relazionali

**RISPOSTA 3**

Lo statement select si compone di diverse istruzioni. C’è un ordine con cui deve essere scritto uno statement select, che sarebbe l’ordine con cui deve essere scritta una query. L’ordine con cui vengono processate le operazioni è diverso.

**(5.) SELECT** consente di indicare i campi, le costanti o le espressioni (anche complesse) del result set

**(1.) FROM** consente di indicare una sorgente dati, cioè la tabella che occorre interrogare

1. **WHERE** consente di indicare condizioni di ricerca, cioè filtra applicati ad una o più colonne della tabella indicata nella FROM
2. **GROUP BY** consente di aggregare i record e di ottenere dei raggruppamenti (come una Pivot)
3. **HAVING** è una condizione di ricerca esattamente come una Where condizion, ma i filtri sono applicati ai raggruppamenti, quindi viene usata solo quando c’è una GROUP BY.
4. **ORDER BY** consente di indicare gli ordinamenti

**RISPOSTA 4**

|  |  |
| --- | --- |
| **TipologieStruttureSport** | |
| **NomeStruttura** | **TipoSport** |
| Struttura1 | Nuoto |
| Struttura2 | Nuoto |
| Struttura3 | Nuoto |
| Struttura4 | Ginnastica Ritmica |
| Struttura5 | Nuoto Sincronizzato |
| Struttura6 | Pallanuoto |
| Struttura7 | Pallanuoto |

Considerando la tabella **TipologieStruttureSport**

attraverso la clausola **GROUP BY** è possibile creare dei raggruppamenti univoci.

A seguito dell'applicazione GROUP BY è consentito utilizzare gli operatori di aggregazione (ad esempio COUNT(), MAX(),...)

**ESEMPIO**

Se volessi contare quante strutture ci sono per lo sport del Nuoto dovrei, raggruppare in maniera univoca per TipoSport e contare quante strutture ci sono per quel raggruppamento:

**SELECT** TipoSport, **COUNT**(NomeStruttura) **AS** ConteggioStrutture

**FROM** TipologieiStruttureSport

**GROUP BY** TipoSport

|  |  |
| --- | --- |
| **Result set** | |
| **TipoSport** | **ConteggioStrutture** |
| Ginnastica Ritmica | 1 |
| Nuoto | 3 |
| Nuoto Sincronizzato | 1 |
| Pallanuoto | 2 |
|  |  |

**RISPOSTA 5**

I vari utenti aziendali nelle loro operazioni quotidiane utilizzano dei sistemi software (ERP, CRM…) che consentono di inserire transazioni.

Queste operazioni sono dette operazioni CRUD (Create, Read, Update, Delete) e sono memorizzate in una basedati.

La base dati che accoglie le transazioni degli utenti è una base dati di tipo **OLTP (Online TransactionProcessing)**. È una base dati progettata per ottimizzare le operazioni CRUD di pochi record e garantisce la consistenza dei dati. Questo perchè implementa una serie di vincoli che controllano l’inserimento di record duplicati, oppure record che devono rispettare le chiavi logiche. È una base dati estremamente normalizzata, nel senso che I dati di una stressa entità vengono distribuiti su più tabelle per ridurre la ridondanza e garantire l’integrità referenziale degli stessi. Le tabelle devono essere organizzate in maniera tale da avere relazioni da 1 a molti ( 1🡪n, ad un record di una tabella ne corrispondo n in un’altra), per poter gestire correttamente le relazioni.

Lo schema OLTP si differenzia dallo schema **OLAP (Online Analytical Processing)**, il quale è denormalizzato. È un database relazionale in cui I dati di una stessa entità sono impacchettati in un’unica tabella. OLAP è il cuore del **Datawarehouse** che è la basedati ottimizzata per l’analisi del dato, per ottimizzare l’interrogazione di una grossa mole di dati… che è quell oche fa la BI.

Lo **STAR SCHEMA**, definisce come progettare un **DWH**. Sostanzialmente questo approccio esplicita che per costruire un database ottimizzato per la gestione del dato occorre individuare due macro-tipologie di tabelle: le tabelle dei “fatti” ( il fatto è un qualsiasi evento aziendale che bisogna analizzare, quantificare e che si evolve nel tempo) e le tabelle delle “dimensioni” ( la tabella dimensionale è una prospettiva di analisi, un punto di vista rispetto al quale filtrare il fatto).

Le **fasi di progettazione dello schema OLTP** sono:

1. Schema concettuale (modellazione E/R a un alto livello di astrazione)
2. Schema logico (progettazione delle tabelle)
3. Schema fisico (creazione delle tabelle nella basedati SQL)

**RISPOSTA 6**

Molte query innestate possono essere scritte come Join e viceversa.

Occorre però tener presente che nel result sets che deriva da uno statement select dove è presente una query innestata non possono essere mostrate colonne della query interna.

Pertanto, a parità di scenario:

* se nel result set è necessario esporre le colonne delle tabelle presenti nella Subquery, è necessario ricorrere alla JOIN;
* se nel result set non è necessario esporre le colonne delle tabelle presenti nella Subquery, è possible utilizzare entrambi gli approcci ed ottenere lo stesso output.

A livello di performance occorre chiedersi quante JOIN è necessario fare per avere un determinato result set. Nel caso in cui dovessero essere “troppe” potrebbe risultare più efficiente utilizzare una Subquery.

**RISPOSTA 7**

DDL e DML sono due macro-tipologie di istruzioni SQL.

* DDL: Data Definition Language (CREATE, ALTER, DROP). Sono istruzioni che consentono di creare un oggetto (le tabelle, …), modificare un oggetto ed eliminare un oggetto.
* DML: Data Manipulation Language (SELECT, INSERT, UPDATE, DELETE) sono tutte quelle istruzioni che consentono di manipolare un dato.

**RISPOSTA 8**

YEAR(Date)

Ad esempio considerando la seguente tabella

|  |  |
| --- | --- |
| **Acquisti** | |
| **IdOrdine** | **DataOrdine** |
| 1 | 22/01/2024 |
| 2 | 23/01/2024 |
| 3 | 24/01/2024 |

Per estrarre l’anno dell’ordine uno, l’istruzione è la seguente

SELECT IdOrdine, YEAR(DataOrdine) AS AnnoOrdine

FROM Acquisti

WHERE IdOrdine = 1

|  |  |
| --- | --- |
| **ResultSet** | |
| **IdOrdine** | **AnnoOrdine** |
| 1 | 2024 |
|  |  |

**RISPOSTA 9**

AND restituisce TRUE quando entrambe le espressioni sono TRUE;

OR restituisce TRUE quando una delle due condizioni è TRUE

**RISPOSTA 10**

Si. In generale la query può essere innestata in qualsiasi punto dello statement select

**RISPOSTA 11**

IN consente di fare un elenco di espressioni in cui individuare una corrispondenza

OR restituisce TRUE quando una delle due condizioni è TRUE. Per ottenere lo stesso risultato che si ha utilizzando l’operatore logico IN, si dovrebbero utilizzare molteplici OR.

**RISPOSTA 12**

Si. BETWEEN restituisce TRUE quando i valori controllati sono >= e <= del range specificato, pertanto include gli estremi.

**RISPOSTA 13**

Quando in una tabella è presente una colonna i cui valori sono i figli di un'altra colonna della tabella stessa, tipicamente c’è una self Join, cioè la tabella è autoreferenziata, nel senso che c'è una gerarchia.

Esempio: considerando la seguente tabella

|  |  |  |
| --- | --- | --- |
| **Clothes** | | |
| **ClothesID** | **ParentClothesID** | **Name** |
| 1 | null | Maglia |
| 2 | null | Pantaloni |
| 3 | null | Scarpe |
| 4 | 1 | T-shirt |
| 5 | 1 | Maglia |
| 6 | 2 | Shorts |
| 7 | 3 | Sneakers |
| 8 | 3 | Decoltè |

È necessario “sdoppiare” la tabella mediante l’utilizzo degli alias, in modo da avere una tabella che ci consente di analizzare i figli (che ha cardinalità n) ed una tabella che ci consente di analizzare il padre ( che ha cardinalità 1)

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

SELECT C1.ClothesID, C1.Name AS NomeProdotto, C2.ParentClothesID, C2.Name AS Category

FROM

Clothes AS C1

JOIN Clothes AS C2

ON C1.ParentClothesID = C2.ParentClothesID

**Task 1: Proponi una progettazione concettuale e logica della base dati**

Progettazione concettuale

Immagine che contiene diagramma, testo, Piano, linea

Descrizione generata automaticamente

Progettazione logica

Immagine che contiene testo, schermata, numero, linea

Descrizione generata automaticamente